

# Payment Management System - Code Logic Walkthrough

by: Nikil Saini([nikils@zeta.tech](mailto:nikils@zeta.tech))

## Project Overview

The Payment Management System is a **Vue 3-based web application** designed to manage users and payments with role-based access. It features:

- Admin: Manage users, view all payments, update payment statuses, and delete users.
- User: Create payments, view personal transaction history.

The app includes a **dashboard with dynamic statistics and interactive tables** for both users and payments.

---

## Project Logic

### 1. Data Handling

- All data is stored in **mock JSON files** (`src/mock-data/data.json`) and **served using JSON Server**.
- Axios is used for **CRUD operations** to fetch, create, update, and delete users/payments.
- For example, deleting a user triggers an HTTP DELETE request to the JSON Server.

### 2. State Management

- **Vuex** is used to manage global state (`userRole` and `isLoggedIn`) across components.
- **Local storage persists logged-in user data**, allowing dashboard rendering based on role.

### 3. Role-Based Access

- **Computed properties** and conditional rendering ensure only authorized users can access certain actions.
- Admin users see extra columns/buttons for managing users and payments.

### 4. Dashboard and UI Logic

- Dashboard cards calculate totals (`totalUsers`, `totalPayments`, `success/pending/failed`) dynamically.
- Payments table supports filtering by status.
- **Components are modular and reusable** (`UserList`, `UserForm`, `DashboardComponent`, `NavbarComponent`).

### 5. Routing

- **Vue Router** handles navigation between dashboard, users, and payments pages.
- Guards and conditional rendering enforce role-based restrictions.

---

## GitHub Repository

[Payment Management System Repository](#)

---

## Instructions to Run the Project

### 1. Clone the repository

```
git clone
https://github.com/nikilsaini89/Payments-Management-System.git
cd Payments-Management-System
```

### 2. Install dependencies

```
npm install
```

3. **Start JSON Server** (serves users and payments data at <http://localhost:3000>)

```
npx json-server --watch src/mock-data/data.json --port 3000
```

4. **Start Vue Development Server**

```
npm run serve
```

5. **Access the app at:**

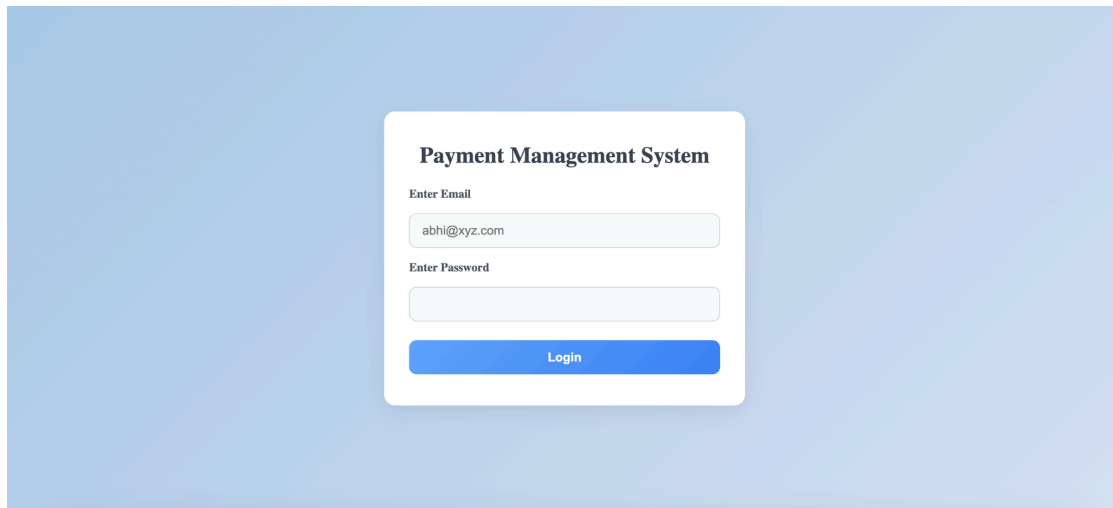
```
http://localhost:8080
```

6. **Login using predefined Admin or User credentials to explore role-based features.**

- **Admin Access:**
  - Email: [nikilsaini74@gmail.com](mailto:nikilsaini74@gmail.com)
  - Password: nikil123
- **User Access:**
  - Email: [sumit.negi@gmail.com](mailto:sumit.negi@gmail.com)
  - Password: sumit123

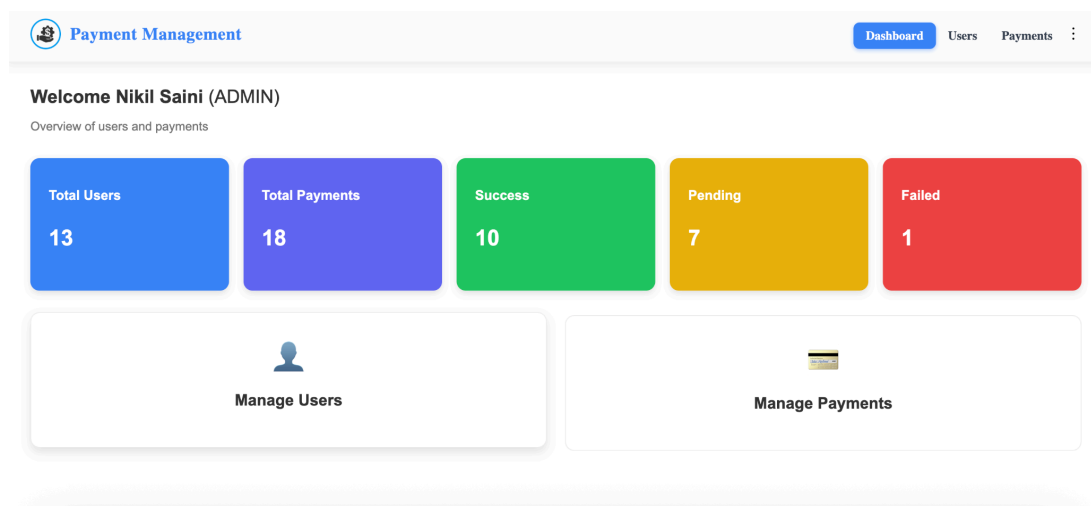
# Project UI

- **Login Page:** A secure login interface for Admins and Users to access their respective dashboards.




The login page features a light blue gradient background. In the center is a white card with rounded corners. The card has a title 'Payment Management System' in bold. Below the title are two input fields: 'Enter Email' with the placeholder 'abhi@xyz.com' and 'Enter Password'. At the bottom of the card is a blue 'Login' button.

- **Admin Dashboard:** Displays user management options and payment statistics for all users at a glance.




The admin dashboard has a light gray header. On the left is the 'Payment Management' logo. On the right are navigation links: 'Dashboard' (active), 'Users', 'Payments', and a menu icon. Below the header, it says 'Welcome Nikil Saini (ADMIN)' and 'Overview of users and payments'. The main content area has five colored cards: 'Total Users' (13, blue), 'Total Payments' (18, purple), 'Success' (10, green), 'Pending' (7, yellow), and 'Failed' (1, red). At the bottom are two white cards: 'Manage Users' with a user icon and 'Manage Payments' with a credit card icon.


- **Admin Users List:** Shows all users with the ability to delete any user directly.

 <b>Payment Management</b>		Dashboard	Users	Payments	⋮
<b>User Management</b>					<a href="#">+ Add User</a>
ID	Name	Email	UPI Id	Actions	
2	Isha Verma	isha.verma@mail.com	isha@upi	<a href="#">Delete</a>	
3	Rohan Mehta	rohan.mehta@mail.com	rohan@upi	<a href="#">Delete</a>	
4	Priya Nair	priya.nair@mail.com	priya@upi	<a href="#">Delete</a>	
5	Aditya Singh	aditya.singh@mail.com	aditya@upi	<a href="#">Delete</a>	
6	Neha Kapoor	neha.kapoor@mail.com	neha@upi	<a href="#">Delete</a>	
7	Karan Malhotra	karan.malhotra@mail.com	karan@upi	<a href="#">Delete</a>	
8	Sanya Gupta	sanya.gupta@mail.com	sanya@upi	<a href="#">Delete</a>	
9	Vikram Rao	vikram.rao@mail.com	vikram@upi	<a href="#">Delete</a>	

- **Admin Payment List:** Lists all payments with an option to update a payment, redirecting to update page on click Update Payment CTA.

 <b>Payment Management</b>		Dashboard	Users	Payments	⋮
<b>Payments List</b>					Select Filter: <span>All</span>
ID	From User (UPI ID)	To User (UPI ID)	Payment Status	Amount	Actions
1	Isha Verma (isha@upi)	Rohan Mehta (rohan@upi)	PENDING	Rs. 750	<a href="#">Update Payment</a>
2	Rohan Mehta (rohan@upi)	Priya Nair (priya@upi)	SUCCESS	Rs. 2000	<a href="#">Update Payment</a>
3	Priya Nair (priya@upi)	Aditya Singh (aditya@upi)	SUCCESS	Rs. 950	<a href="#">Update Payment</a>
4	Aditya Singh (aditya@upi)	Neha Kapoor (neha@upi)	SUCCESS	Rs. 500	<a href="#">Update Payment</a>
5	Neha Kapoor (neha@upi)	Karan Malhotra (karan@upi)	PENDING	Rs. 1800	<a href="#">Update Payment</a>
6	Karan Malhotra (karan@upi)	Sanya Gupta (sanya@upi)	SUCCESS	Rs. 3000	<a href="#">Update Payment</a>
7	Sanya Gupta (sanya@upi)	Vikram Rao (vikram@upi)	FAILED	Rs. 1450	<a href="#">Update Payment</a>
8	Vikram Rao (vikram@upi)	Meera Iyer (meera@upi)	SUCCESS	Rs. 2200	<a href="#">Update Payment</a>

- **Payment Detail Page:** Shows detailed information about a selected payment.


 Payment Management

Dashboard Payments

Payment Details

ID:	216
From :	sumit@upi
To:	karan@upi
Amount:	₹456
Status:	PENDING
Method:	UPI
Reference ID:	TXN738832
Timestamp:	19/09/2025, 10:01:00

- **Change Payment Status:** Admin can modify the status of any payment (Pending, Success, Failed).

 Payment Management

Dashboard Users Payments

Edit Payment

From UPI ID

isha@upi

To UPI ID

rohan@upi

Amount

750

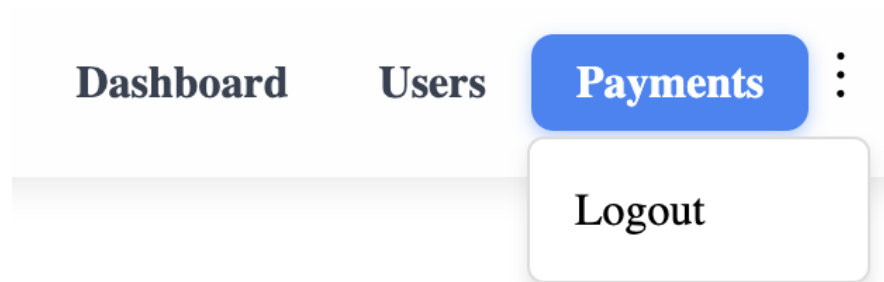
Status

Pending

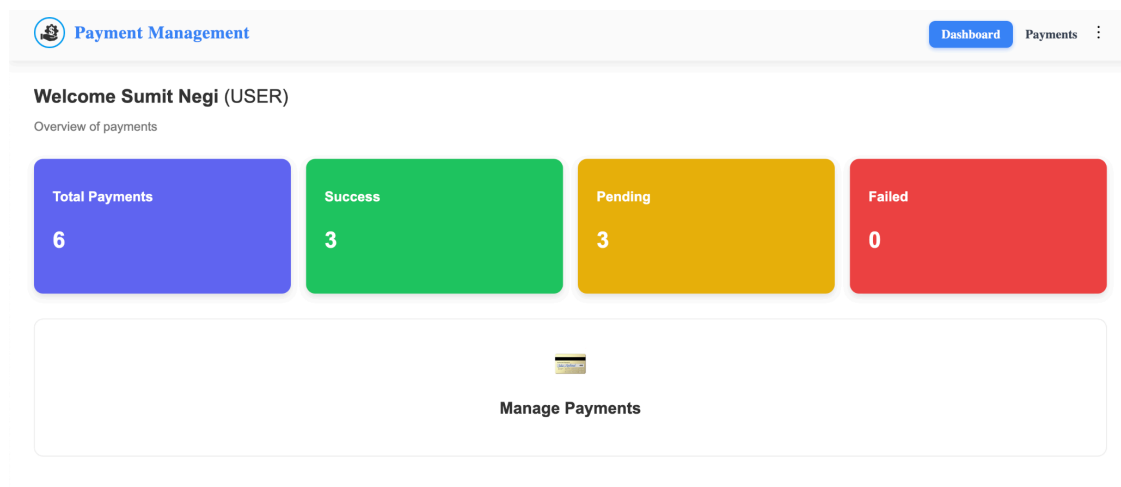
Update

Cancel


- **Logout Functionality:** Enables both Admins and Users to securely log out from the application.



- **User Dashboard:** Provides a personalized view limited to the user's own payments and statistics.



- **User Payment List:** Similar to the admin view but without the option to update payments.

 Payment Management


Dashboard **Payments** ⋮

Payments List

Select Filter: All **+ Create Payment**

ID	From User (Upi ID)	To User (Upi ID)	Payment Status	Amount
1	Sumit Negi (sumit@upi)	Neha Kapoor (neha@upi)	SUCCESS	Rs. 34
2	Sumit Negi (sumit@upi)	Sanya Gupta (sanya@upi)	PENDING	Rs. 34
3	Sumit Negi (sumit@upi)	Karan Malhotra (karan@upi)	PENDING	Rs. 456
4	Sumit Negi (sumit@upi)	Sanya Gupta (sanya@upi)	SUCCESS	Rs. 54
5	Sumit Negi (sumit@upi)	Sanya Gupta (sanya@upi)	SUCCESS	Rs. 80000
6	Sumit Negi (sumit@upi)	Priya Nair (priya@upi)	PENDING	Rs. 7

- **Create Payment:** Allows Users to initiate a new payment directly from their dashboard.

 Payment Management

Dashboard **Payments** ⋮

Add New Payment

To UPI ID

-- Select User --

Amount

Enter amount


Create

Cancel



## Error States:

- **No payments found** for the current user or selected filter.

 Payment Management


Dashboard **Payments** ⋮

**Payments List**

Select Filter: **Failed** ⌵ **+ Create Payment**

ID	From User (Upi ID)	To User (Upi ID)	Payment Status	Amount
No Payments to show!				

- **Missing required fields** while creating a payment.

 Payment Management

Dashboard **Payments** ⋮

### Add New Payment

**To UPI ID**

-- Select User -- ⌵

User is required

**Amount**

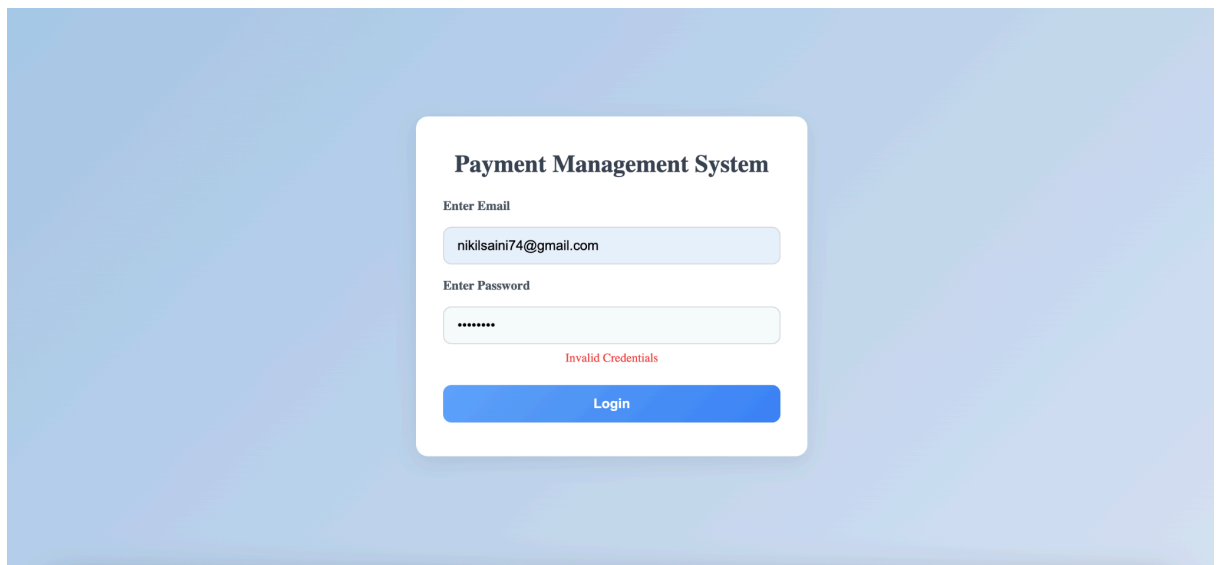
Enter amount

Amount must be greater than 0

**Create** **Cancel**

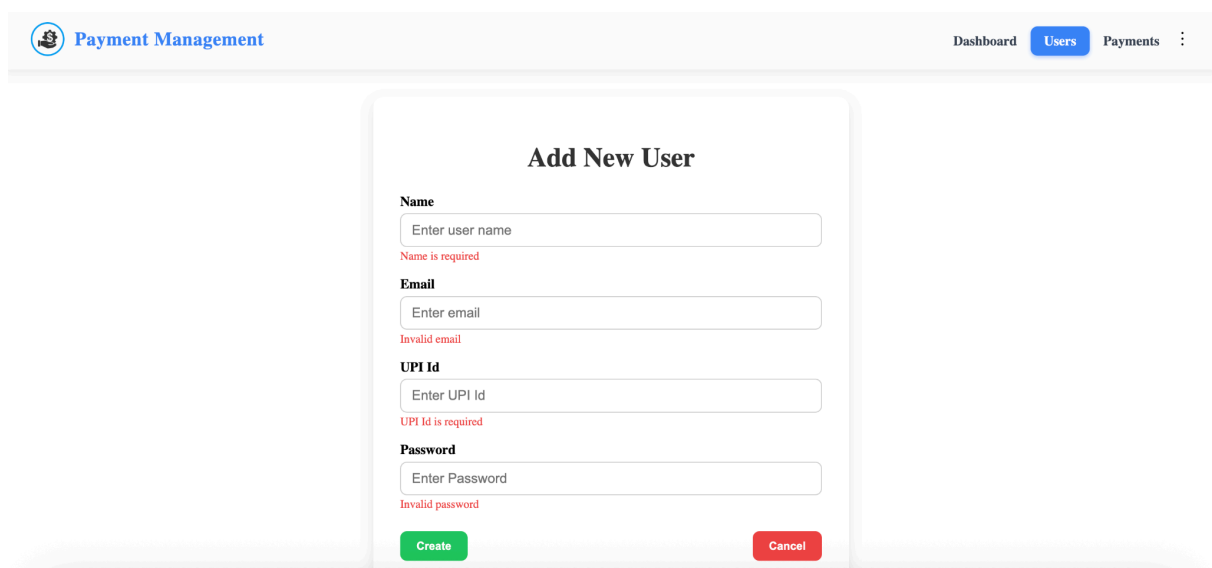
localhost:8080

- **Invalid credentials** provided during login.



The image shows a login form for a "Payment Management System". The form is centered on a light blue background. It has two input fields: "Enter Email" with the value "nikilsaini74@gmail.com" and "Enter Password" with masked characters "\*\*\*\*\*". Below the password field, a red error message "Invalid Credentials" is displayed. At the bottom of the form is a blue "Login" button.

- **Missing required fields** while creating a user.



The image shows a web application interface for "Payment Management". The header includes a logo, the title "Payment Management", and navigation links for "Dashboard", "Users" (which is highlighted), and "Payments". The main content area displays a form titled "Add New User". The form has four sections, each with a label, an input field, and a red error message below it: "Name" (input: "Enter user name", error: "Name is required"), "Email" (input: "Enter email", error: "Invalid email"), "UPI Id" (input: "Enter UPI Id", error: "UPI Id is required"), and "Password" (input: "Enter Password", error: "Invalid password"). At the bottom of the form are two buttons: a green "Create" button and a red "Cancel" button.